

---

# **shaderdef Documentation**

***Release 0.8.3***

**Nicholas Bishop**

August 30, 2016



<b>1</b>	<b>Example program</b>	<b>3</b>
<b>2</b>	<b>shaderdef.shader module</b>	<b>5</b>
<b>3</b>	<b>shaderdef.glsl_var module</b>	<b>7</b>
<b>4</b>	<b>shaderdef.interface module</b>	<b>9</b>
<b>5</b>	<b>shaderdef.glsl_funcs module</b>	<b>11</b>
<b>6</b>	<b>shaderdef.glsl_types module</b>	<b>13</b>
<b>7</b>	<b>Indices and tables</b>	<b>15</b>
	<b>Python Module Index</b>	<b>17</b>



Transform Python code into GLSL shaders.

Website: <https://github.com/nicholasbishop/shaderdef>



---

## Example program

---

```
#!/usr/bin/env python3

"""This demo shows how to define a very simple shader program with a
vertex shader and a fragment shader.

The vertex shader is defined in the `vert_shader` function. Note that
the inputs and outputs are annotated; this is how shaderdef knows
what types to use in the generated GLSL code.

Inputs and outputs are grouped together using a Python class. For
example, this shader program's vertex attributes are defined in
`VsIn`. It inherits from `AttributeBlock` to mark its members as vertex
attributes. (There's a UniformBlock for declaring uniform inputs.)

Shader outputs are set using the `return` keyword. The return type
should be a class such as `VsOut`; pass the outputs as keyword
arguments.
"""

from shaderdef import (AttributeBlock, FragmentShaderOutputBlock,
                       ShaderDef, ShaderInterface)

from shaderdef.glsl_types import vec2, vec4

class VsIn(AttributeBlock):
    position = vec2()

class VsOut(ShaderInterface):
    gl_position = vec4()

class FsOut(FragmentShaderOutputBlock):
    color = vec4()

def vert_shader(attr: VsIn) -> VsOut:
    return VsOut(gl_position=vec4(-attr.position.x, attr.position.y, 1.0, 1.0))

def frag_shader() -> FsOut:
    return FsOut(color=vec4(1.0, 0.0, 0.0, 1.0))

def print_shaders():
    sdef = ShaderDef(vert_shader=vert_shader, frag_shader=frag_shader)
    sdef.translate()
```

```
print('\nvertex shader:')
print('-----')
print(sdef.vert_shader)
print('\nfragment shader:')
print('-----')
print(sdef.frag_shader)

def main():
    """
    >>> print_shaders() # doctest: +NORMALIZE_WHITESPACE
    vertex shader:
    -----
    #version 330 core
    layout(location=0) in vec2 position;
    void main() {
        gl_Position = vec4(-attr.position.x, attr.position.y, 1.0, 1.0);
    }
    fragment shader:
    -----
    #version 330 core
    layout(location=0) out vec4 color;
    void main() {
        color = vec4(1.0, 0.0, 0.0, 1.0);
    }
    """
    print_shaders()

if __name__ == '__main__':
    main()
```



---

## shaderdef.shader module

---

**class** `shaderdef.shader.ShaderDef` (*vert\_shader*, *frag\_shader*, *geom\_shader=None*)

Bases: `object`

**add\_function** (*function*)

Add a utility *function* to the shader program.

Each utility function is currently emitted in all shader stages regardless of which stage or stages the function is actually used in.

**frag\_shader**

Get the GLSL code for the fragment shader.

**geom\_shader**

Get the GLSL code for the geometry shader.

**get\_uniforms** ()

**translate** ()

**vert\_shader**

Get the GLSL code for the vertex shader.



---

## shaderdef.glsl\_var module

---

**class** shaderdef.glsl\_var.**GlslVar** (*name*, *gtype*, *interpolation=None*)

Bases: object

Represent a GLSL variable declaration (or struct member).

**declare** ()

**declare\_attribute** (*location=None*)

**declare\_output** (*location=None*)

**declare\_uniform** ()

**gtype** = Attribute(name='gtype', default=NOTHING, validator=None, repr=True, cmp=True, hash=True, init=True, con

**interpolation** = Attribute(name='interpolation', default=None, validator=None, repr=True, cmp=True, hash=True, i

**name** = Attribute(name='name', default=NOTHING, validator=None, repr=True, cmp=True, hash=True, init=True, conv

shaderdef.glsl\_var.**location\_str** (*location*)



---

## shaderdef.interface module

---

```
class shaderdef.interface.AttributeBlock (**kwargs)
    Bases: shaderdef.interface.ShaderInterface
        classmethod declare_input_block (instance_name=None, array=None)
class shaderdef.interface.FragmentShaderOutputBlock (**kwargs)
    Bases: shaderdef.interface.ShaderInterface
        classmethod declare_output_block (array=None)
class shaderdef.interface.GLGsIn (**kwargs)
    Bases: shaderdef.interface.ShaderInterface
        gl_position
class shaderdef.interface.ShaderInterface (**kwargs)
    Bases: object
        classmethod block_name ()
        classmethod declare_input_block (instance_name, array=None)
        classmethod declare_output_block (array=None)
        classmethod get_vars ()
        classmethod instance_name ()
class shaderdef.interface.UniformBlock (**kwargs)
    Bases: shaderdef.interface.ShaderInterface
        classmethod declare_input_block (instance_name, array=None)
shaderdef.interface.snake_case (string)
```



---

## shaderdef.glsl\_funcs module

---

```
shaderdef.glsl_funcs.end_primitive()  
shaderdef.glsl_funcs.exp2(var)  
shaderdef.glsl_funcs.geom_shader_meta(input_primitive, output_primitive, max_vertices)  
shaderdef.glsl_funcs.length(vec_type)  
shaderdef.glsl_funcs.mod(num1, num2)
```





---

## shaderdef.glsl\_types module

---

```
shaderdef.glsl_types.Array1  
    alias of GlslArray  
shaderdef.glsl_types.Array10  
    alias of GlslArray  
shaderdef.glsl_types.Array11  
    alias of GlslArray  
shaderdef.glsl_types.Array12  
    alias of GlslArray  
shaderdef.glsl_types.Array13  
    alias of GlslArray  
shaderdef.glsl_types.Array14  
    alias of GlslArray  
shaderdef.glsl_types.Array15  
    alias of GlslArray  
shaderdef.glsl_types.Array16  
    alias of GlslArray  
shaderdef.glsl_types.Array2  
    alias of GlslArray  
shaderdef.glsl_types.Array3  
    alias of GlslArray  
shaderdef.glsl_types.Array4  
    alias of GlslArray  
shaderdef.glsl_types.Array5  
    alias of GlslArray  
shaderdef.glsl_types.Array6  
    alias of GlslArray  
shaderdef.glsl_types.Array7  
    alias of GlslArray  
shaderdef.glsl_types.Array8  
    alias of GlslArray  
shaderdef.glsl_types.Array9  
    alias of GlslArray
```

```

class shaderdef.glsl_types.ArraySpec(element_type, length)
    Bases: object

    Represents an array declaration.

    This type isn't currently intended to be used by client code directly, it's just a convenient form for internal use.

    element_type = Attribute(name='element_type', default=NOTHING, validator=None, repr=True, cmp=True, hash=True, init=True)

    classmethod from_ast_node (node)
        Create a GlslArray from an AST node if possible.

        If the node cannot be converted then None is returned.

    length = Attribute(name='length', default=NOTHING, validator=None, repr=True, cmp=True, hash=True, init=True, c
class shaderdef.glsl_types.GlslArray(gtype)
    Bases: typing.Generic

class shaderdef.glsl_types.GlslType(*args, **kwargs)
    Bases: typing.SupportsAbs, typing.SupportsInt, typing.SupportsFloat

shaderdef.glsl_types.mat2
    alias of GlslType

shaderdef.glsl_types.mat3
    alias of GlslType

shaderdef.glsl_types.mat4
    alias of GlslType

class shaderdef.glsl_types.noperspective
    Bases: object

shaderdef.glsl_types.vec2
    alias of GlslType

shaderdef.glsl_types.vec3
    alias of GlslType

shaderdef.glsl_types.vec4
    alias of GlslType

class shaderdef.glsl_types.void
    Bases: object

```

---

## Indices and tables

---

- `genindex`
- `modindex`
- `search`



## S

`shaderdef.glsl_funcs`, [11](#)  
`shaderdef.glsl_types`, [13](#)  
`shaderdef.glsl_var`, [7](#)  
`shaderdef.interface`, [9](#)  
`shaderdef.shader`, [5](#)



## A

`add_function()` (`shaderdef.shader.ShaderDef` method), 5  
`Array1` (in module `shaderdef.glsl_types`), 13  
`Array10` (in module `shaderdef.glsl_types`), 13  
`Array11` (in module `shaderdef.glsl_types`), 13  
`Array12` (in module `shaderdef.glsl_types`), 13  
`Array13` (in module `shaderdef.glsl_types`), 13  
`Array14` (in module `shaderdef.glsl_types`), 13  
`Array15` (in module `shaderdef.glsl_types`), 13  
`Array16` (in module `shaderdef.glsl_types`), 13  
`Array2` (in module `shaderdef.glsl_types`), 13  
`Array3` (in module `shaderdef.glsl_types`), 13  
`Array4` (in module `shaderdef.glsl_types`), 13  
`Array5` (in module `shaderdef.glsl_types`), 13  
`Array6` (in module `shaderdef.glsl_types`), 13  
`Array7` (in module `shaderdef.glsl_types`), 13  
`Array8` (in module `shaderdef.glsl_types`), 13  
`Array9` (in module `shaderdef.glsl_types`), 13  
`ArraySpec` (class in `shaderdef.glsl_types`), 13  
`AttributeBlock` (class in `shaderdef.interface`), 9

## B

`block_name()` (`shaderdef.interface.ShaderInterface` class method), 9

## D

`declare()` (`shaderdef.glsl_var.GlslVar` method), 7  
`declare_attribute()` (`shaderdef.glsl_var.GlslVar` method), 7  
`declare_input_block()` (`shaderdef.interface.AttributeBlock` class method), 9  
`declare_input_block()` (`shaderdef.interface.ShaderInterface` class method), 9  
`declare_input_block()` (`shaderdef.interface.UniformBlock` class method), 9  
`declare_output()` (`shaderdef.glsl_var.GlslVar` method), 7  
`declare_output_block()` (`shaderdef.interface.FragmentShaderOutputBlock` class method), 9  
`declare_output_block()` (`shaderdef.interface.ShaderInterface` class method), 9  
`declare_uniform()` (`shaderdef.glsl_var.GlslVar` method), 7

## E

`element_type` (`shaderdef.glsl_types.ArraySpec` attribute), 14  
`end_primitive()` (in module `shaderdef.glsl_funcs`), 11  
`exp2()` (in module `shaderdef.glsl_funcs`), 11

## F

`frag_shader` (`shaderdef.shader.ShaderDef` attribute), 5  
`FragmentShaderOutputBlock` (class in `shaderdef.interface`), 9  
`from_ast_node()` (`shaderdef.glsl_types.ArraySpec` class method), 14

## G

`geom_shader` (`shaderdef.shader.ShaderDef` attribute), 5  
`geom_shader_meta()` (in module `shaderdef.glsl_funcs`), 11  
`get_uniforms()` (`shaderdef.shader.ShaderDef` method), 5  
`get_vars()` (`shaderdef.interface.ShaderInterface` class method), 9  
`gl_position` (`shaderdef.interface.GIGsIn` attribute), 9  
`GIGsIn` (class in `shaderdef.interface`), 9  
`GlslArray` (class in `shaderdef.glsl_types`), 14  
`GlslType` (class in `shaderdef.glsl_types`), 14  
`GlslVar` (class in `shaderdef.glsl_var`), 7  
`gtype` (`shaderdef.glsl_var.GlslVar` attribute), 7

## I

`instance_name()` (`shaderdef.interface.ShaderInterface` class method), 9  
`interpolation` (`shaderdef.glsl_var.GlslVar` attribute), 7

## L

`length` (`shaderdef.glsl_types.ArraySpec` attribute), 14  
`length()` (in module `shaderdef.glsl_funcs`), 11  
`location_str()` (in module `shaderdef.glsl_var`), 7

## M

`mat2` (in module `shaderdef.glsl_types`), 14  
`mat3` (in module `shaderdef.glsl_types`), 14

mat4 (in module shaderdef.gslsl\_types), 14  
mod() (in module shaderdef.gslsl\_funcs), 11

## N

name (shaderdef.gslsl\_var.GslslVar attribute), 7  
noperspective (class in shaderdef.gslsl\_types), 14

## S

ShaderDef (class in shaderdef.shader), 5  
shaderdef.gslsl\_funcs (module), 11  
shaderdef.gslsl\_types (module), 13  
shaderdef.gslsl\_var (module), 7  
shaderdef.interface (module), 9  
shaderdef.shader (module), 5  
ShaderInterface (class in shaderdef.interface), 9  
snake\_case() (in module shaderdef.interface), 9

## T

translate() (shaderdef.shader.ShaderDef method), 5

## U

UniformBlock (class in shaderdef.interface), 9

## V

vec2 (in module shaderdef.gslsl\_types), 14  
vec3 (in module shaderdef.gslsl\_types), 14  
vec4 (in module shaderdef.gslsl\_types), 14  
vert\_shader (shaderdef.shader.ShaderDef attribute), 5  
void (class in shaderdef.gslsl\_types), 14